Technical Foundations for Emerging Battlefield Information Management
(or, Why Naming Objects in Computers Is So Important)

Dr. Sam Chamberlain*
U.S. Army Research Laboratory
ATTN: AMSRL-IS-TP
Aberdeen proving Ground, MD 21005-5067
wildman@arl.mil

Abstract

Sometimes we overlook the most fundamental issues of battle command, especially when automation is involved. This is due to our natural instincts to automate manual procedures rather than exploit the true capabilities of new technologies. This paper addresses one of these cases—the identification and management of many entities inside of many interconnected computers. It is argued that this problem strikes at the heart of battle command automation process and, consequently, the development and execution of mission capability packages (MCP). This paper has three main theses: (1) the formal concept of organization forms the backbone on which all other battle command entities and functions relate; (2) databases have to be fast, reliable, and unambiguous to be useful to, and across, military applications; and (3) every item inside a computer must be universally "named" and any identification strategy must be globally unique and consistent. Together, these three tenets provide a unifying information structure on which affordable applications can be built, regardless of country, service, or branch/ department affiliation. However, one must not be fooled into believing that this is merely a computer science problem; it is primarily a military science problem with some computer science technology "sprinkled in."

* This research has been sponsored by the DOD Command and Control Research Program under the ASD(C3I).

1. The Organization Backbone

Military databases are filled with numerous different entities. These include representations of both physical items (like equipment and personnel) and conceptual items (like plans and organizations). Although there are several definitions for "organization" or "unit," for example,

> Webster's: *organization* – an administrative structure with a mission[1], or

> Joint Services Dictionary: *unit* – (DOD, NATO)
> 1. Any military element whose structure is prescribed by competent authority, … specifically, part of an organization[2],

a rigorous, formal definition is required to address the myriad of issues involved when one automates a battle command process (like MCP

[1] *Webster's 7th Collegiate Dictionary*
[2] http://www.dtic.mil/doctrine/jel/doddict

development). Over the past several years, formal data modeling tools and processes have permeated the database development community. A notable example is the relational IDEF1X information modeling technique [Bruce, 1992]. However, even though the "business rule" approach has provided a tremendous improvement to the data modeling process, there are still many procedural issues left undefined that need to be addressed. Logic programming will help to define some of this information [Grant and Minker, 1992; Robinson, 1992], but basic military science is required to first describe what it is one does during the "task organizing" function in the military.

It is important to understand that an organization is a virtual entity; that is, one can not touch an organization. In its simplest state, it is mental clustering of real-world objects, collected together based upon human thoughts. It typically manifests itself as "organization charts," computer listings, or database entries that describe people and equipment "assigned" to the organization. In this sense, the concept of an organization forms the basis of command. Before one can command, there must be something to command, thus, organizations are formed, be they formal or informal. But the rules for building these structures are often elusive when the formal definition process begins (as the author has found from discussions with military experts). It is easy to build structures that, although meaningful to a human, are not so apparent when they must be represented inside a computer. Consider the following example.

A few years ago the Navy announced that it was going to reorganize its operational force into "permanent core battle groups." The following text is taken from the electronic versions of the Navy News Service (NNS) announcements:

> NORFOLK, Va. (NNS) – The Atlantic Fleet's surface combatant ships are being reorganized into six core battle groups, nine destroyer squadrons and a new Western Hemisphere Group. … Once reorganization is completed, two cruisers will be permanently assigned to each carrier battle group. At the start of the intermediate training phase, a four-ship destroyer squadron, two submarines, and a replenishment ship will join the core group to establish the battle group. … When the transition period is complete, the following ship assignments will apply: …
>
> Cruiser-Destroyer Group Two/
> George Washington Battle Group
> - USS George Washington
> - USS South Carolina
> - USS Normandy …[3]
>
> WASHINGTON (NNS) – Pacific Fleet reorganization underway. (This is the second of a two-part series highlighting the reorganization of the Atlantic and Pacific fleets.) The Pacific Fleet's surface ships are being reorganized into six core battle groups and eight destroyer squadrons. … *Permanent core battle groups* will *include* a battle group commander, aircraft carrier, carrier air wing and at least two cruisers. …
>
> Commander Cruiser-Destroyer Group Five/
> USS Kitty Hawk Battle Group
> - USS Kitty Hawk
> - USS Antietam
> - USS Cowpens …[4]

To Navy experts, the intent of this reorganization is quite obvious. But when this structure is represented formally inside a computer database, it is not as straightforward as the text suggests.

[3] http://www.chinfo.navy.mil/navpalib/news/navnews/nns95/nns95031.txt (NNS455)

[4] http://www.chinfo.navy.mil/navpalib/news/navnews/nns95/nns95032.txt (NNS470)

For example, how should a permanent operational organization be represented? Is a permanent carrier battle group in the same category as a permanent Unified Command? Are they truly permanent, that is, when they are not deployed, do they still exist? If so, to whom do they belong when they are not deployed? If not, then are they considered a new organization each time they are stood up for deployment, or are they actually just a representation of a habitual relationship (i.e., not permanent after all)? The point is that even this simple case presents interesting questions that have to be handled consistently if one is to automate the process of building MCPs or any task organization. There must be specific criteria, or at least guidance, defined to unambiguously determine when the creation of a new organization is warranted or when it is sufficient to restructure existing organizations. Otherwise, there will not be a consistent semantics (i.e., meaning) to the concept of "organization."

A related issue is: who is allowed to make this determination? At one extreme one may argue that only Congress can "create" organizations. For example, Congress decides the size of the Army and how many divisions it may have. A brigade commander can not suddenly decide to create a new battalion. However, one may build an Army battalion task force without creating any new organizations. In this case, existing organizations are merely rearranged and given new aliases. For example, companies may be exchanged between battalions via attachments. No new people are hired and no new equipment is procured. Only the command assignments are temporarily altered, and often, a new nickname is applied.

Similarly, the Marines have predefined "slots" under the Marine Expeditionary Unit (MEU) organization to which existing ground, air, and support organizations are attached. Consequently, a new organization is not created for each MEU deployment; instead, different existing organizations are task-organized (e.g., a battalion into a Battalion Landing Team) and attached into an existing slots.

On the other hand, is a Joint Task Force (JTF) a new organization? In this case, it may depend on the situation. The phrase used in the DOD Dictionary definition is a "force that is *constituted* and so *designated* by the Secretary of Defense, CINCs, etc. …" which can mean either situation.[5] So, it appears that a JTF can be either a new organization, or it can be a reinforced existing organization that is given a new "nickname." The criteria for this needs to be defined.

Defining the relationships between organizations and, in particular, the "parent-child" relationship is a major part of the definition process for organizations. Once an organization's existence is determined, children organizations (or sub-units) must be linked to the parent organization. This "task organization" function is a key step in the planning process, and information technology must provide the capability to do this quickly and easily.

Simply stated, organizations are hierarchical structures and there are many alternative precepts by which to construct them. If one were to ask 10 people to draw an organization chart of the same organization, one could easily get back 10 different charts based upon the perspective or the job of the person drawing the chart. Part of the reason for this is the difference between *administrative* and *operational* organizations and their chains of command. The Navy makes very clear distinctions between these structures; however, in the Army, the differences begin to blur as one moves down the echelons (e.g., at company level and below). For example, Army Tables of Organization and

[5] http://www.dtic.mil/doctrine/jel/doddic

Equipment (TOE) show the administrative organization from a logistics and personnel perspective (normally, at the company level). Organization charts derived from TOEs provide a different picture than charts derived from the perspective of an operational chain of command. A good example is a U.S. Army Mechanized Infantry Platoon equipped with M2 Bradley Fighting Vehicles (BFV). The TOE structure does not begin to describe the true organization structure employed in the field.

Another interesting situation is the strong temptation to use task organizational structure (however it is defined) as the basis for other, unrelated uses. One example is to use organizational names as the basis of assigning network host names to battlefield computers. Obviously, this is for the convenience of users since the computers do not rely on hostnames (they use Internet Protocol [IP] addresses). Inconsistencies in the organizational definition make this a dubious practice, especially when task organizing is a common occurrence and organizational parent-child relationships are in constant flux. Hostnames (and IP addresses) can change radically as organizations move between their administrative and operational structures. Ultimately, in a truly digitized force, no one should have to know a hostname to communicate with a known organization. All that should be needed is the name of the organization, or better yet, an icon of the organization.

Another important issue is the scope of the organization concept. If one builds an organization chart (a hierarchical tree structure) the leaves of the tree eventually end up as individual people. Should the term organization extend down to the individual level? Certainly there is nothing that precludes this, and from an information management perspective, this provides a consistent, general approach to the structure. A significant advantage of extending the concept of organization down to the individual level is that it facilitates the unification of the administrative (i.e., logistics and personnel) and operational organization perspectives. For example, in the current Army systems, personnel are assigned to "slots" that are not hierarchically defined but, rather, fall under a common, flat organization, typically a company. If the organization structure is carried down below company level (i.e., to platoons, squads, sections/fire teams, etc.), then slots from a personnel perspective merge one-to-one with organizations from an operational perspective. This allows a common representation between these two database schemas.

Once the organization tree structure is defined, equipment (and other assets) can also be aligned with the structure. Unlike people, that exist at the leaves of the tree, equipment may be aligned at any level (or echelon). For example, although an organization chart for a Navy ship may extend down to the individual sailor, the ship's hull can be linked to an organization that is an ancestor of, or encapsulates, the entire crew (e.g., it could be called the "USS XYZ Crew"). The children of this organization would be the ship departments and so forth. If the ship sinks, the organization would still be in effect, even though the sailors are floating in the water. In an Army mechanized infantry example, a BFV could be linked with an organization called "BFV-2/A Section/ 3rd Platoon." The children of this organization would be organizations called "Bradley Commander," "Gunner," and "Driver" that correspond to individual soldier slots. In an Air Force example, a single-pilot aircraft (like an F-15C) may have a one-to-one correspondence between an organization, person (i.e., the pilot), and the equipment (i.e., the aircraft), while a large aircraft (e.g., an AWACS) could be aligned with an organization with several levels of structure below the umbrella "crew" organization. The approach is completely general with the details of the structure left to each organization that is authorized to define the structure (e.g., force development groups).

Another interesting question is: Do organizations have to contain people? Robotics technology is moving at a fast pace. If a person can be an organization, can an autonomous system, like an unmanned aerial vehicle (UAV) be an organization? Would a "platoon" of UAVs be an organization? This is a definition question because the proposed approach does not preclude this from occurring.

These questions indicate that, although “we all know” what an organization is, there are few formal definitions of the processes and the criteria used to create, modify, reorganize, and delete organizations. This is precisely the reason that this problem must be formally addressed and debated. The thesis of this paper is that the concept of an organization (or unit, or whatever term is deemed most appropriate) forms the backbone by which all other military concepts are related. A formal definition of organization is paramount to combining together the myriad concepts that make up battlefield management. Once the structure is built, personnel can be aligned with the leaves of the tree, equipment can be aligned anywhere in the tree, and conceptual entities like plans, target lists, coordination measures, communication networks, and a myriad others can be linked together based on the unifying concept of an organization. Even better, these entities can be quickly and unambiguously modified based on changes to the organization.

The point of this discussion is to emphasize that getting a consistent and unambiguous definition for “organization” is one of the basic military science problems that must be resolved before information technology can be consistently applied across battlefield management domains. If this can not be accomplished, then truly difficult tasks, like the unification of command and control with intelligence, will be nearly impossible.

2. The Organization Identifiers

For information management systems to be useful to warriors, they must be fast, reliable, and unambiguous, even under constrained conditions (e.g., due to bandwidth limitations). To accomplish this, one must remember that humans and machines have different strengths. For example, people prefer to identify things using words or pictures, while machines like numbers. This is very apparent in search-intensive processes like those found in information technology.

Tracking “who is who” is one of the basic challenges encountered in a battle command system. Currently, identification of organizations takes many forms. (Some examples are the Unit Identification Code [UIC], the Unit Reference Number [URN], and the Master Unit List [MUL] number.) There is no simple, unified way to identify an organization. Consequently, such a feature would be a significant step toward achieving system interoperability.

Every entity in a database must have a unique (primary) key to identify it; this includes organizations. Often, primary keys are created by combining primary keys from other entities. This is often illustrated using the example of a movie store where individual video tapes are identified with primary keys that are composed of a unique movie name followed by a copy number (e.g., “A Bridge Too Far” – Copy 1). Although the term “naming convention” typically brings to mind human readable forms of creating and identifying unique organization names, this is not necessary for computers. From a data-processing perspective, there is nothing simpler than an integer. *Surrogate keys* are primary keys that have no special meaning; a good example of this is integers. Integers offer several advantages to include simplicity, fast performance, and easily verifiable uniqueness. Normally, users will never know surrogate keys exist, but the database management system and application programs can use them extensively to greatly enhance performance and flexibility.

The recommendation in this paper is to use surrogate keys as the primary keys for organizations. In this case, a simple four-byte integer is recommended called an Organization Identifier, or "org-id." A 32-bit integer has the capability to uniquely enumerate more than 4.3 billion organizations (actually, $2*10^{32}–1$), providing that all the possible numbers are used (i.e., none are wasted). Unfortunately, there is a strong tendency to break up the address space and parcel it out. This is purely to assist in human interpretation and intervention; the machines could care less. Therefore, the second part of the recommendation is to treat the surrogate keys merely as integers, with no special assignment of values to the bits (i.e., if bit x and y are 0s, then it is an Air Force organization). Integer values would be assigned first-come, first-served, with no waste. This has three obvious advantages: (1) it is very general and prevents humans from encoding information into the bits that they will later learn to regret; (2) it provides a very terse manner in which to identify organizations, and (3) it allows the computers to do simple integer operations that are very fast, because all reference to organizations are handled as integers. For example, a complete tree structure can be built simply by including an integer attribute in each organization that references its parent organization (be it the default, current, or some other definition of parent). This allows very quick access up and down the organization tree.

Since humans do not always relate well to numbers, there can always be alternate (primary) keys stored as attributes to the organization entity. For example, there could be a unit name that uses a standard "naming convention" (if someone would like to develop one). However, a basic assumption is that in the future, operators will not be accessing databases directly but, rather, will be using application programs that in turn access databases. Thus, operators will be insulated from details like "surrogate keys."

Administration of this approach could be done with a set of org-id servers. When an authorized user wants to create a new organization, an org-id must be obtained. To do this, a request is sent to one of several org-id servers. In return, probably within some maximum time, the requestor must return to the server at least two pieces of information: (1) the name of the domain to which that org-id has been assigned and (2) some sort of status flag (such as active or dormant). Note that an org-id server is not an organization server. The purpose of an org-id server is to ensure that org-ids are unique and not wasted (hence, the status field). The actual organization information would be located in "organization servers" maintained by the individual services or agencies. For example, the org-id server would keep track of the fact that org-id 123456789 is an active org-id that belongs to the "army.mil" domain. To find detailed information about organization 123456789, one would contact one of the army.mil organization servers.

However, more information could be stored at the org-id servers. For example, to maintain an authoritative tree structure of the current known DOD organizations, two other attributes could be maintained: (1) a character string for the name of the organization and (2) the org-id of the organization's administrative parent organization. This would allow a skeleton tree structure to be built that could assist with validation and error checking. There are numerous other options that could be applied here, but most do not fall under the purview of an org-id server but, rather, the organization servers.

The previous discussion of organization issues may now be more apparent. The issues surrounding org-id *stability* have significant implementation ramifications. At this point, the reader may have a vision of a huge bottleneck being created as thousands of users flood an org-id server as they attempt to create new organizations during a battle. A basis assertion in this paper is that org-ids are very stable. This means that in nearly all cases, building a new task organization does not require the generation of new org-ids. Only special cases, like the creation of a JTF, may require an ad hoc visit to the org-id server (even this can be circumvented by pre-allocating a few org-ids to the unified commands). Creating the stable set of "functionally static" org-ids is primarily an administrative function, akin to the building of TOEs.

In the vast majority of cases, the building of a new task organization simply requires that existing organizations, with stable org-ids, be temporarily associated with new parent organizations. In other words, new operational structures are built with *existing* organizations. In an Army example, two battalions may switch companies to build a battalion task force with a temporary name of "Task Force Alpha." In a Navy example, a specific destroyer squadron, two submarines, and a supply ship are temporarily attached to a core carrier battle group. (Recall that a ship's crew is represented by an umbrella organization.) In a Marine example, a battalion landing team is constructed and temporarily attached to an MEU. In an Air Force example, a strike package is created by combining together four existing aircraft (each represented by an organization). The details of this example are still being debated.

The assertion in the previous examples is that none of the task organization examples required the creation of a new organization but, rather, only the restructuring of the links to existing organizations. Although the associations are dynamic, the existence of the individual organizations is static.

3. The General Naming Problem

Ultimately, one wants to uniquely identify all the entities distributed across the battle space via myriad computers. Imagine that the primary keys of all entities are surrogate keys, that is, they are integers. This is in stark contrast to the usual method of defining primary keys by building them from numerous other primary keys (as in IDEF1X data modeling). Now, the size of a primary key could be a constant based on a "large integer" (e.g., in the 64 to 128 bit range).

The task is now to guarantee that no two integers are reused, at least within a given timeframe. This is where org-ids help. Conceptually, every organization is allowed to create new entities and distribute them to other databases. When a new entity is created, the database on which it is created assigns a surrogate key that is the concatenation of the org-id of the organization that controls the database and another integer. All the database management system must do is ensure that it never uses the same integer twice, at least within a specific timeframe. The size of the integer can be fixed or variable. If the second integer is sized at 4 bytes, then this gives every organization the ability to create over 4.3 billion entities before the numbers wrap around. If this is not big enough, then an expanding system can be implement in which the first bit indicates if another 4-byte integer follows. This allows unlimited chaining and tremendous flexibility to the implementers.

Although the purpose of the surrogate key is only to provide a unique identifier (i.e., one that is guaranteed universally unique), this approach has a secondary benefit because one always knows what organization created the entity.

Whether this secondary feature should be exploited is debatable because using this information violates the basic tenet that a surrogate key should be meaningless. Abusing this policy can easily lead to problems because unanticipated conditions may arise in the future that invalidate the assumptions that were used when meaning was added to the keys.

In this approach, a surrogate key is a machine-generated primary database key that is forever bound to a data item at the time of its creation. Therefore, an entity's surrogate key (or object ID) is an integral part of it and moves with it as it traverses numerous computers. In other words, the surrogate key that is assigned when it is created stays with the entity as it propagates among battle command systems.

Fortunately, much of the data in a battle command system is reference material (e.g., the range of an F-15 with wing tanks, the existing ships within the Navy, or on a larger scale, the existing organizations within the DOD). In general, reference material is relatively static and, consequently, much of it can be predefined and, as required, preloaded into computer databases to include the associated surrogate keys. For example, the information stored in the Communications-Electronic Operating Instructions (CEOI) can be preloaded. This includes standard networks (like a particular company command network), subscribers to the network, default or habitual connections, and the equipment used to communicate. All this information is relatively static, and only the attributes change (i.e., frequencies, call signs, cryptographic codes, etc.). Thus, the information can be preloaded with common surrogate keys across battlefield computers. Changes can be simple updates in predefined attributes of existing entities that can be transmitted tersely between hosts or broadcast from satellites to numerous hosts.

Thus, surrogate keys can be used to identify common reference material across machines; this is regardless of whether the database is in a French, English, German, or Turkish computer system, or whether the database is in a relational, object-oriented, or other form. Entities with the same surrogate key are semantically equivalent. This greatly improves potential interoperability while also significantly reducing bandwidth requirements.

4. Summary

This paper argues that the concept of an organization is the central theme to which all other battle command entities relate. Therefore, formally defining how we build and structure organizations is essential to the automation of battle command processes.

The unique identification of entities is the enabling feature that allows all other battle command functions to be realistically integrated, regardless of differences in language, nation, service, branch, or function. For example, when organization identifiers (org-ids) extend down to the individual person level, personnel system identifiers suddenly match operational system identifiers, thus allowing the potential for the two systems to interoperate via a common context.

A proposed approach is to (1) uniquely identify all organizations with a unique surrogate key called an "organization ID," and then (2) allow the databases at each unit to build a composite surrogate key that is composed of an organization ID combined with a surrogate key generated from the local database. This guarantees that no two organizations will generate identical keys, provided that the organization IDs are unique. This allows large volumes of reference material to be defined ahead of time that can be shared between disparate battlefield systems. If bandwidth is a problem, reference material can be preloaded thus reducing the amount of information that must be transmitted using limited communication resources. If bandwidth is plentiful, this approach allows common reference material to be broadcast to numerous hosts. An incidental feature of this scheme is that one always knows "who" (what organization) created a piece of information

Implementing the assignment of unique organization IDs will require a process similar to what is used thousands of times daily to obtain new IP addresses via network information centers. And, like IP addresses, in the vast majority of cases, once they are assigned, they will remain relatively stable.

Some will say that a plan to provide unique organization IDs (an integer) to all DOD organizations is unrealistic. But they would have said the same thing 30 years ago when someone suggested that every computer in the world would have a unique ID, called an IP address. It is the standardization and acceptance of IP that made the Internet possible, and unique organization IDs can have the same effect on the automation of distributed battle command that IP addresses have had on computer networks.

References

[Bruce, 1992] Thomas A. Bruce. *Designing Quality Databases with IDEF1X Information Models*. Dorset House Publishing, New York, NY, 1992.

[Grant and Minker, 1992] J. Grant and J. Minker. The Impact of Logic Programming on databases. *Communications of the ACM, Vol 35(3)*, March 1992.

[Robinson, 1992] J. Robinson. Logic and Logic Programming. *Communications of the ACM, Vol. 35(3)*, March 1992.